Please replace the paragraph starting at page 1, line 32, and ending at page 2, line 10, with the following replacement paragraph:

A typical personal computer (PC) has one or more local buses such as ISA, VESA, and/or PCI buses for connection of user-selected devices. The PC communicates with the devices using device addresses typically indicated by settings of jumper wires or toggle switches on the device. Problems can arise because there is no guaranty that a set of devices, made by different manufacturers, can operate together without address conflicts. Even if a set of devices can operated together, connection of the devices to a local bus may require that the user identify address conflicts and change address settings to avoid the conflicts. This can make adding devices to a PC difficult.

Please replace the paragraph starting at page 2, line 11, and ending at page 2, line 29, with the following replacement paragraph:

Common devices connected to an ISA bus include serial input/output (I/O) devices such as a printer, a modem, or a mouse. Some operating environments such as Microsoft WINDOWS™ running in conjunction with MS-DOS operating system provide for standardized connections to serial devices coupled to the ISA bus. In particular, WINDOWS™ and MS-DOS support four communication or COM ports, each having a predefined base device address. This allows resolution of device address conflicts if each serial device coupled to the ISA bus has settings for at least four different base device addresses. Each COM port is for connection to a serial device which contains a communication interface known as a Universal Asynchronous Receiver/Transmitter (UART). The UART is well known in the art and described, for example, in the 1994

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

"Telecommunications Data Book" from National Semiconductor Corporation, which is incorporated by reference herein in its entirety.

Please replace the paragraph starting at page 2, line 30, and ending at page 3, line 10, with the following replacement paragraph:

Fig. 1 illustrates conventional communications between a serial device 110 and an application 140 via an operating environment 130 and a communications driver 120. Operating environment 130 provides a library of subroutines which application 140 calls to communicate with serial device 110. The subroutines call communications driver 120 which writes and reads data and control information to and from a UART 105. The standardized communication interface illustrated in Fig. 1 reduces the complexity of application 140 because application 140 is not required to implement a variety of communication protocols. Accordingly, most applications are written for the standard interface. However, a standard hardware UART may be unsuitable or too expensive for some devices. Accordingly, techniques are needed which provide non-standard devices with the benefits of a standard UART interface.

Please replace the paragraph starting at page 3, line 25, and ending at page 4, line 5, with the following replacement paragraph:

In one embodiment of the invention, a computer system includes a non-standard device and a COM driver for the non-standard device. The non-standard device connects to an I/O slot corresponding to a first COM port but has a register set which differs from the standard register set for a UART. The COM driver contains: a UART emulation which in response to a procedure requesting access to a register of a UART at the first COM port,

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

instead accesses storage locations in main memory of the computer system; and an I/O handler which transfers values between the storage locations in main memory and the register set of the device. Optionally, the system includes a standard device having a UART coupled to an I/O slot corresponding to a second COM port, and the COM driver contains routines for accessing the standard device.

Please replace the paragraph starting at page 5, line 29, and ending at page 5, line 36, with the following replacement paragraph:

In one embodiment of the invention, operating environment 130 includes Microsoft WINDOWS™ which supports four COM ports for communications with up to four serial devices connected to an ISA bus 115. A standard COM port occupies a slot of eight addresses on ISA bus 115. The eight addresses correspond to registers of a standard UART, which function as shown in Table 1.

Please replace the paragraph starting at page 7, line 13, and ending at page 7, line 28, with the following replacement paragraph:

During initialization of COM ports, COM driver 220 determines which of the four COM ports are allocated to standard UART devices, determines if a non-standard device is present, and then allocates an unassigned COM port and I/O slot to device 210. Serial device 210 is initially locked during start-up. When locked, serial device 210 receives a data signal DATA and address signal ADDR from ISA bus 115, but does not respond to any address. COM driver 220 unlocks device 210 by transmitting address signal ADDR and data signal DATA with values equal to a predefined pattern recognized by device 210. When unlocked, the base device address of device 210 depends on information that COM driver 220 provides

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

while unlocking device 210. Device 210 replies to the address sent by COM driver 220 to indicate that device 210 is present.

Please replace the paragraph starting at page 8, line 1, and ending at page 8, line 19, with the following replacement paragraph:

Unlocking circuit 300 contains a base address decoder 330 and a pattern generator 310. While the device is locked, base address decoder 330 asserts a signal SEL to pattern generator 310. Pattern generator 310 generates a signal PAT that represents a byte which is from a predefined sequence and corresponds to the value of signal SEL. Signal SEL starts in an initial state, such as indicating a count value of zero or a maximum count. Each time the local bus carries an address signal ADDR having a recognized value, base address decoder 330 compares data signal DATA from the local bus to signal PAT and if signals PAT and DATA are equal, changes signal SEL so that signal SEL advances toward a final state. Otherwise, signal SEL is reset to the initial state. Advancing signal SEL can for example increment a count value from an initial state (minimum value) toward a final state (maximum value) or decrement the count from an initial state (maximum value) to a final state (minimum value).

Please replace the paragraph starting at page 8, line 28, and ending at page 9, line 8, with the following replacement paragraph:

Fig. 3B shows a block diagram of an embodiment of base address decoder 330. Base address decoder 330 contains AND gates 331, 332, and 333 which are coupled to address lines of ISA bus 115. AND gate 333 asserts a signal ADX when signal IOWCN indicates the computer is writing data and address signal ADDR[11:0] has the form 001x 111x 1111 binary

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

where x indicates that bits ADDR4 and ADDR8 are "don't care" bits, i.e. can have either value 0 or 1. The COM driver selects values for bits ADDR8 and ADDR4 so that signal ADDR[11:0] does not correspond to any other device coupled to the ISA bus. A signal AEN indicates when a DMA controller in the computer places an address on ISA bus 115. In the embodiment shown, unlocking circuit 300 does not respond to the DMA controller, and signal ADX is only asserted if signal AEN indicates address signal ADDR[11:0] is not from the DMA controller.

Please replace the paragraph starting at page 10, line 20, and ending at page 10, line 32, with the following replacement paragraph:

Many alternative patterns and pattern generators may be used in place of the embodiment shown in Fig. 3C. For example, pattern generator 310 can be implemented using a memory such as a read-only memory where signal SEL[2:0] is an address signal or implemented using combinatorial logic where signal SEL[2:0] is an input signal. Each value in the pattern can be longer or shorter than a byte and can be a constant value independent of signal SEL. Further, the predetermined pattern can be longer or shorter than five values. Increasing the length of the pattern reduces the chance of a device being unintentionally locked.

Please replace the paragraph starting at page 11, line 13, and ending at page 11, line 25, with the following replacement paragraph:

Signals PCA4 and PCA8 are latched when an address signal ADDR[11:0] is asserted for a byte following the predefined pattern. The byte following the predefined pattern does not match signal PAT[7:0] from pattern generator 310. Accordingly, counter 335 is reset to

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979